Amendments to the Claims:

A clean version of the entire set of pending claims, including amendments thereto, is submitted herewith per 37 CFR 1.121(c)(3). This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1-20. (Canceled)

21. (Original) A microcontroller that supports a plurality of message objects, comprising:

a processor core that runs applications;

a module that processes incoming messages;

data memory including a first memory segment that provides a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object; and,

a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

22. (Original) The microcontroller as set forth in claim 21, wherein the incoming messages include multi-frame, fragmented messages, and the module automatically assembles the multi-frame, fragmented messages.

23. (Original) The microcontroller as set forth in claim 21, wherein the module includes the memory-mapped registers.

24. (Original) The microcontroller as set forth in claim 21, wherein the processor core, the module, and the memory interface unit are contained on a single integrated circuit chip.

25. (Original) The microcontroller as set forth in claim 24, wherein the first and second memory segments are contained on the integrated circuit chip.

26. (Original) The microcontroller as set forth in claim 24, wherein the memory interface unit includes two independent arbiters.

27. (Original) The microcontroller as set forth in claim 21, wherein the memory interface unit arbitrates access according to an alternate winner policy, wherein a previous loser is designated a current winner.

28. (Original) A microcontroller that supports a plurality of message objects, comprising:

a processor core that runs applications;

a module that processes incoming messages, wherein the processor core and the module are contained on a single integrated circuit chip;

data memory including a first memory space that is located on the integrated circuit chip and a second memory space that is located off the integrated circuit chip, the first memory space including a first memory segment that provides at least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object

containing respective command/control fields for configuration and setup of that message object; and,

a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory spaces, that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the second memory space and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the second memory space or to the same one of the first and second memory segments.

29. (Original) The microcontroller as set forth in claim 28, wherein the incoming messages include multi-frame, fragmented messages, and the module automatically assembles the multi-frame, fragmented messages.

30. (Original) The microcontroller as set forth in claim 28, wherein the module includes the memory-mapped registers.

31. (Original) The microcontroller as set forth in claim 28, wherein the memory interface unit is contained on the single integrated circuit chip.

32. (Original) The microcontroller as set forth in claim 28, wherein the second memory space provides at least a portion of the message buffer memory space.

33. (Original) The microcontroller as set forth in claim 28, wherein the memory interface unit includes two independent arbiters dedicated to a respective one of the first and second memory spaces.

34. (Original) The microcontroller as set forth in claim 28, wherein the memory interface unit arbitrates access according to an alternate winner policy, wherein a previous loser is designated a current winner.

35. (Currently Amended) A method for operating a microcontroller that supports a plurality of message objects, the ~~CAN~~ microcontroller including a processor core that runs applications[[,]] and a module that processes incoming messages, wherein the processor and module are included in a same integrated circuit chip, and further including a data memory including a first memory space that is located on the integrated circuit chip and a second memory space that is located off the integrated circuit chip, the first memory space including a first memory segment that provides at least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object, the method comprising:

permitting the processor core and the module to concurrently access a different respective one of the first and second memory segments; and,

arbitrating access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

36. (Original) The method as set forth in claim 35, wherein the arbitrating access step is performed in accordance with an alternate winner policy, wherein a previous loser is designated a current winner.

37. (Currently Amended) The method as set forth in claim 36, wherein the arbitrating step is performed by a memory interface unit contained in the microcontroller.

38. (Currently Amended) A method for operating a microcontroller that supports a plurality of message objects, the microcontroller including a processor core that runs applications, a module that processes incoming messages, and a data memory including a first memory space that is located on an integrated circuit chip on which the microcontroller and the module are incorporated, and a second memory space that is located off the integrated circuit chip, the first memory space including a first memory segment that provides at least a portion of a message buffer memory space that includes a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object, the method comprising:

permitting the processor core and the module to concurrently access a different respective one of the first and second memory spaces;

permitting the processor core and the module to concurrently access a different respective one of the first and second memory segments;

arbitrating access to the second memory space when the processor core and the module request concurrent access to the second memory space; and,

arbitrating access to the same one of the first and second memory segments when the processor core and the ~~CAN/CAL~~ module request concurrent access to the first and second memory segments.

39. (Original) The method as set forth in claim 38, wherein the arbitrating access step is performed in accordance with an alternate winner policy, wherein a previous loser is designated a current winner.

40. (Original) The method as set forth in claim 39, wherein the arbitrating step is performed by a memory interface unit contained in the microcontroller.

41. (Original) A bus station comprising a microcontroller that supports a plurality of message objects, comprising:

a processor core that runs applications;

a module that processes incoming messages;

data memory including a first memory segment that provides a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object; and,

a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the same one of the first and second memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.

42. (Original) A bus system comprising a microcontroller that supports a plurality of message objects, comprising:

a processor core that runs applications;

a module that processes incoming messages;

data memory including a first memory segment that provides a plurality of message buffers associated with respective ones of the message objects, and a second memory segment that provides a plurality of memory-mapped registers for each of the message objects, the memory-mapped registers for each message object containing respective command/control fields for configuration and setup of that message object; and,

a memory interface unit that permits the processor core and the module to concurrently access a different respective one of the first and second memory segments, and that arbitrates access to the same one of the first and second

memory segments when the processor core and the module request concurrent access to the same one of the first and second memory segments.